Intelligent Information Systems 9999
ISBN 666-666-666, pages 1–9

Applying grade methods to detect similarity of semantic categories of nouns for semantic valence dictionary creation

Elżbieta Hajnicz and Marek Wiech

Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract

The paper describes the process of finding the similarity of 26 top-most wordnet *semantic categories of nouns* in particular syntactic slots for a selected set of verbs. The long-range goal is to extend a syntactic valence dictionary for Polish verbs with semantic information (represented by lists of *semantic categories*). An aggregation of similar *categories* will protect the dictionary from an unnecessary proliferation of dictionary entries. We use grade data exploration methods to find similarity between *semantic categories.* First we perform grade analysis for each syntactic slot separately, next we combine the obtained information in one matrix. All visualisations were prepared in application GradeStat.

1 Introduction

It turns out that understanding a syntactic structure of texts is insufficient to obtain satisfactory results in any Natural Language Processing (NLP) task, such as *machine translation*, *information extraction* and *retrieval*, *question answering* etc. Actually, semantic information is indispensable.

In practical applications focused on specific domains (e.g., medicine, finance, sport) such information is gathered in very popular *ontologies.* On the other hand, more universal lexical semantic resources, such as wordnets (**??**) and FrameNet (**??**) are created.

Our ultimate goal is to extend a syntactic valence dictionary for Polish verbs with semantic information, represented by means of 26 wordnet top-most *semantic categories* of nouns. In particular, each verb scheme will be composed of a list of corresponding syntactic slots equipped with list of categories. Syntactic slots we discuss cover noun phrases and prepositional phrases, while semantic categories represent meaning of words (here, nouns) occurring in sentences on the corresponding positions. However, creating a separate dictionary entry for every tuple of pairs ⟨*syntactic slot*, *semantic category*⟩ possible for a particular verb scheme would cause an unnecessary proliferation of entries. Moreover, most of these entries would describe the verb with the same meaning. It is obvious that our goal is to represent every meaning of each verb as a singular entry and different meanings as separate entries, i.e., we want to detect polysemy.

In order to achieve this goal we need some kind of similarity measure of semantic categories. In this paper we propose a way of applying *grade methods* (**?**) to perform this task.

2 Data resources

The first resource we have applied in our work is the Polish WordNet, called *Słowosieć* (**??**). It is a network of lexical-semantic relations, an electronic thesaurus with a structure modelled on that of the Princeton WordNet (**?**) and those constructed in the EuroWordNet (**?**) project. Polish WordNet describes the meaning of a lexical unit of one or more words by placing this unit in a network of links which represent such relations as synonymy, hypernymy, meronymy, etc. For the present work we do not use the whole structure of the net, but only a set of 26 predefined semantic categories (listed e.g. on Fig. 2) located on the top-most level of the actual hierarchy. Using these categories, 7815 nouns (most frequent in the balanced subcorpus of the IPI PAN Corpus) were classified.

The second resource has been the IPI PAN Corpus (**??**), from which we have selected a small subcorpus containing 165 253 simple sentences for 99 selected verbs. This subcorpus has been parsed with the metamorphic grammar *Świgra* (**?**), each parse reduced to its flat form identifying only the top-most phrases. Next, reduced parse forests for each sentence was disambiguated by means of an EM selection algorithm (**??**). After these operations the number of sentences decreased to 41 793.

Resultant reduced parses were augmented with semantic categories of their semantic heads. A version of EM selection algorithm was used to disambiguate these categories (as some nouns have more than one meaning) (**??**). The final form of the sentence representation is shown in the following example:

```
% 'Ona nie wzięła się z twardych reguł wolnego rynku.'
   (She/It hasn't emerged from hard rules of the free market.)
<wziąć :np:nom:  :prepnp:z:gen:  :sie:>
0-9  wziąć    neg:fin:sg:f:ter::
              [0-1:np:on:sg:nom:f:ter::  pronoun,
              1-4:sie,
              4-9:prepnp:z:reguła:gen::  cognition]

% 'Prezydent spotkał się na rynku z mieszkańcami Bochni.'
   (The president has met on the market place with citizens of Bochnia.)
<spotkać :np:nom:  :prepnp:na:loc:  :sie:>
0-8  spotkać  aff:fin:sg:m:ter::
              [0-1:np:prezydent:sg:nom:m1:ter::  person
              1-3:sie,
              3-8:prepnp:na:rynek:loc::  location]
0-8  spotkać  aff:fin:sg:m:ter::
              [0-1:np:prezydent:sg:nom:m1:ter::  person,
              1-3:sie,
              3-8:prepnp:na:bochnia:loc::  location]
```

All occurrences of semantic categories are counted w.r.t. verbs and syntactic slots they appear with in reduced parses of 41 793 sentences we have. Therefore, we obtain a 3D matrix: slots × verbs × categories, represented as a set of matrices verbs × categories for each slot. Unfortunately, it is possible that—even after disambiguation—we deal with more than one reduced parse per sentence and more than one category per slot. Thus, we count categories proportionally to parses and slots they appear with. Hence our counts need not be integer.

Our goal is in a way opposite to the popular task of *Word Sense Disambiguation* (WSD), as we start from a collection of sentences with disambiguated semantic categories of nouns. Contrary, we want to aggregate semantic categories of nouns in such a way that the interpretation of verb arguments will be correct. For instance in sentences,

> *Piotr*$_{\textsf{person}}$ *przejechał park*$_{\textsf{location}}$ *brata samochodem*$_{\textsf{artifact}}$.
> (*Piotr cross his brother's park in a car.*)
> *Piotr*$_{\textsf{person}}$ *przejechał psa*$_{\textsf{animal}}$ *brata samochodem*$_{\textsf{artifact}}$.
> (*Piotr run over his brother's dog by a car.*)

we have different meanings of the verb *przejechać*, hence we want to have two different entries for it in valence dictionary, with location and animal on the object position, correspondingly. On the other hand, in sentences:

> *Piotr*$_{\textsf{person}}$ *kupił bratu*$_{\textsf{person}}$ *park*$_{\textsf{location}}$. (*Piotr bought his brother a park.*)
> *Piotr*$_{\textsf{person}}$ *kupił bratu*$_{\textsf{person}}$ *psa*$_{\textsf{animal}}$. (*Piotr bought his brother a dog.*)

we deal with the same meaning of the verb, and we want to have one entry for it.

Our idea is based on an assumption that we have a space (1 or more D) defined by means of similarity measure over 26 categories of nouns. We want to aggregate semantic categories appearing for a particular verb in a particular slot (represented by a row in a corresponding matrix). We perform this by detecting connected regions in the space of categories. Thus, the categories animal and location are supposed to land in one region for *buying* and two separate regions for *cross / run over* objects (accusative case slot). The reason is that the corresponding rows of acc matrix differ in a way enabling such conclusions: we can buy almost everything material, and we can cross / run over only some different precisely determined things.

To obtain this, we need a similarity measure of these 26 categories. However, the methods used in WSD (**?**), in particular, for automatic thesaurus (**?**) and other semantic dictionaries construction cannot be applied here, as they apply similarity measures between words to determine much more fine-grained concepts. This concerns also the work of **?** (**?**), who consider syntactic and semantic dependencies between verbs and their arguments. Nevertheless, they do not consider the relations on the top-level of wordnet hierarchy as we plan.

3 Finding similarity measure between noun categories

Necessary compactness of this article does not allow us explaining all technical details concerning finding similarity measure, therefore we will skim through it and refer to suitable papers. In short, we linearly ordered noun categories for every slot and put orderings (represented by introduced here *grade regression values*) in a new matrix slots × categories (which we call *the final matrix*).

From the 41 793 sentences 31 matrices verbs × categories were extracted, which each matrix corresponding to particular syntactic slot (cf. Fig. 2, listing in rows all 31 slots). Each matrix has the same structure: there are 167 selected verbs in rows and 26 noun categories in columns. At the intersection of *verb* and *noun category* there is a frequency of how many times a particular noun category (in a particular slot) has appeared in conjunction with a particular verb, therefore each matrix of *verbs* × *noun category* is a contingency table.

One of the most important grade methods is the Grade Correspondence Analysis (called in short "GCA"), an algorithm that tends to find permutations of rows and columns of a data matrix for which a given grade dependence or regularity measure Spearman's $\rho*$ (or Kendall's τ) becomes maximal. The Spearman $\rho*$ for probability table P with m rows, k columns, where p_{is} is the probability of ith row in sth column in this table, is defined as:

$$\rho^*(P) = 3\sum_{i=1}^{m}\sum_{s=1}^{k}(p_{is}(2Sc_{row}(i)-1)(2Sc_{col}(s)-1))$$

$$Sc_{row}(i) = (\sum_{j=1}^{i-1} p_{j*}) + \frac{1}{2}p_{i*}, \qquad Sc_{col}(s) = (\sum_{t=1}^{s-1} p_{*t}) + \frac{1}{2}p_{*s}$$

where marginal sums $p_{j*} = \sum_{s=1}^{k} p_{js}$ and $p_{*t} = \sum_{t=1}^{m} p_{ts}$.

The formal definitions and friendly explanations are given in (**???**). A fine example (in Polish) of implementing grade methods to linguistics data is in (**?**); the algorithm is implemented in program GradeStat[1].

The most important fact here is that by only permuting rows and columns by GCA we increase "regularity" inside the matrix, and very often receive ordering which can be interpreted by an analyst. Figure 1 shows an example matrix, here it is *verbs* × *noun categories* for *nominative* case. The Figure shows two overrepresentation maps[2]. The left map has rows and columns ordered alphabetically, while the right one has rows and columns ordered by GCA.

The right map seems to be highly regular. We can say that some *latent traits* are "governing" the ordering of rows and of columns, and that GCA has revealed a trend in data. Moreover, each column (*noun category*) and each row (*verb*) has a value of *grade regression* function assigned[3]. Formally:

[1] The application *GradeStat*, implementing all grade algorithms is being developed at ICS PAS; the program is available to download at `http://gradestat.ipipan.waw.pl`, and has interface switchable between Polish and English.

[2] An overrepresentation map is a chart in which the intensity of the rectangle on the intersection of row (particular verb) and of column (noun category) shows if the corresponding frequency is smaller than expected (white colour), almost as expected (grey) or higher than expected (black). Width of the columns and of the rows depends on their size, for example column *person* has very high frequencies, so the width of this column is proportional to the sum of all frequencies of verbs in noun semantic category *person* (it is more than half of all frequencies in this matrix).

[3] Values of g.r.f. belong to $[0, 1]$; 0.5 means that column is not monotone dependent with others.

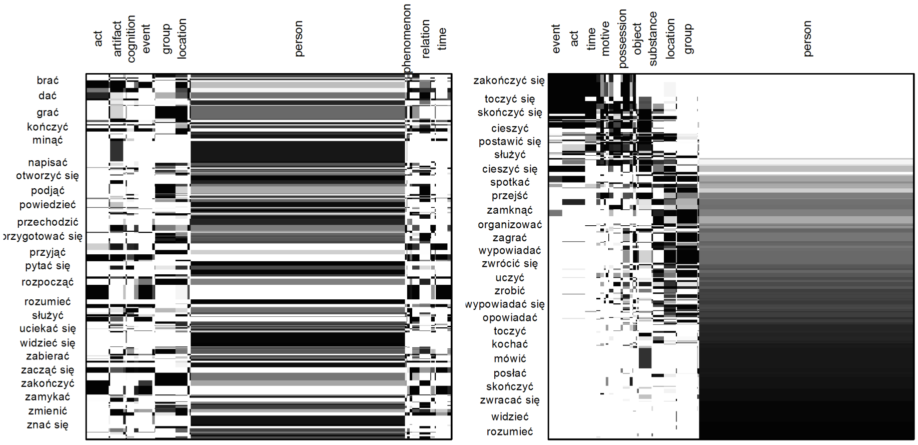


FIGURE 1: Overrepresentation map for matrix *verbs* × *noun categories* for *nominative* case slot; left map: rows and columns ordered alphabetically; right map: matrix ordered by GCA; some row and column names were omitted to improve clarity

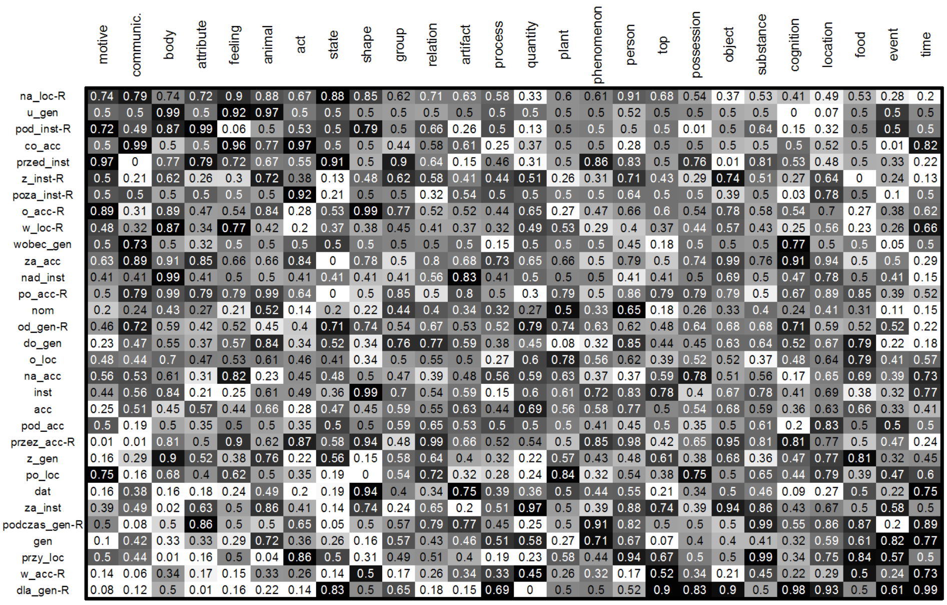
$$Regr_{col}(s) = \frac{\sum_{i=1}^{m}(p_{is}Sc_{row}(i))}{p_{*s}}$$

We perform GCA on each of verbs × categories matrices and then record *ranks* and values of *grade regression* of every semantic category of nouns. The values were counted starting from the left side, so the first column received *rank* 1 and the last one receive *rank* 26; *grade regression* may have values from 0 to 1, so the first column from the left received the smallest value and the last the highest. If the noun category does not appear for particular slot (i.e. has zeros for each verb) it is convenient to put 0.5 as its *grade regression value.* As later analyses have shown that ranks were performing poorly in determining similarity between noun categories, we concentrate only on *grade regression* values.

We take a *grade regression* value for each column (representing *category*) from every verbs × categories matrix. Putting this information together we obtained matrix with 31 rows and 26 columns, where each row corresponds to syntactic slot and columns correspond to noun categories.

GCA always gives us not just one matrix with optimally ordered rows and columns but a pair of them: the found one and its symmetrically reversed matrix (**?**, p. 269). In other words matrix with columns ordered col_A, col_B, col_C and rows ordered row_1, row_2, row_3, row_4 has the same value of grade dependence measure (Spearman's $\rho*$ or Kendall's τ) as the matrix with columns ordered col_C, col_B, col_A and rows ordered row_4, row_3, row_2, row_1. It is not a problem when we analyse only one matrix and are interested in having it ordered optimally, so it does not matter if we analyse original matrix or its symmetrically reversed copy. But in the case of comparison of optimal orderings of *noun categories* in the set of 31 different *slots* our choices should agree, i.e., the orderings should be chosen so that the resulting set of optimally ordered columns is least differentiated. To achieve this goal we have to extend our final $m \times k$ matrix to $(2m) \times k$ matrix by adding 31 *reversed* rows, perform GCA on the matrix and select only upper or lower half of the matrix (as in this case GCA divides the matrix into two symmetric parts, we can select any of them, upper of lower). We chose the upper half and again, for the third time, we used GCA to reveal a possible latent trend in it. The resultant final matrix is shown in Fig. 2 (the rows with reversed ordering have "-R" suffix added to their names).

The final matrix contains information on which position was every *noun category* for each syntactic slot. Therefore, we can for example say that for syntactic slot *nom* categories event and act are very similar, because they have close values of grade regression (0.11 and 0.14), hence they are adjacent in matrix presetned on Fig. 1. Unfortunately, the map on Fig. 2 is weakly regular, since expected regularity concerns categories for particular syntactic slots (i.e., it manifests in single rows), not the whole matrix.



	motive	communic.	body	attribute	feeling	animal	act	state	shape	group	relation	artifact	process	quantity	plant	phenomenon	person	top	possession	object	substance	cognition	location	food	event	time
na_loc-R	0.74	0.79	0.74	0.72	0.9	0.88	0.67	0.88	0.85	0.62	0.71	0.63	0.58	0.33	0.6	0.61	0.91	0.68	0.54	0.37	0.53	0.41	0.49	0.53	0.28	0.2
u_gen	0.5	0.5	0.99	0.5	0.92	0.97	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.52	0.5	0.5	0.5	0.5	0	0.07	0.5	0.5	0.5
pod_inst-R	0.72	0.49	0.87	0.99	0.06	0.5	0.53	0.5	0.79	0.5	0.66	0.26	0.5	0.13	0.5	0.5	0.5	0.5	0.01	0.5	0.64	0.15	0.32	0.5	0.5	0.5
co_acc	0.5	0.99	0.5	0.5	0.96	0.77	0.97	0.5	0.5	0.44	0.58	0.61	0.25	0.37	0.5	0.5	0.28	0.5	0.5	0.5	0.5	0.5	0.52	0.5	0.01	0.82
przed_inst	0.97	0	0.77	0.79	0.72	0.67	0.55	0.91	0.5	0.9	0.64	0.15	0.46	0.31	0.5	0.86	0.83	0.5	0.76	0.01	0.81	0.53	0.48	0.5	0.33	0.22
z_inst-R	0.5	0.21	0.62	0.26	0.3	0.72	0.38	0.13	0.48	0.62	0.58	0.41	0.44	0.51	0.26	0.31	0.71	0.43	0.29	0.74	0.51	0.27	0.64	0	0.24	0.13
poza_inst-R	0.5	0.5	0.5	0.5	0.5	0.5	0.92	0.21	0.5	0.5	0.32	0.54	0.5	0.5	0.5	0.5	0.64	0.5	0.5	0.39	0.5	0.03	0.78	0.5	0.1	0.5
o_acc-R	0.89	0.31	0.89	0.47	0.54	0.84	0.28	0.53	0.99	0.77	0.52	0.52	0.44	0.65	0.27	0.47	0.66	0.6	0.54	0.78	0.58	0.54	0.7	0.27	0.38	0.62
w_loc-R	0.48	0.32	0.87	0.34	0.77	0.42	0.2	0.37	0.38	0.45	0.41	0.37	0.32	0.49	0.53	0.29	0.4	0.37	0.44	0.57	0.43	0.25	0.56	0.23	0.26	0.66
wobec_gen	0.5	0.73	0.5	0.32	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.15	0.5	0.5	0.5	0.45	0.18	0.5	0.5	0.5	0.77	0.5	0.5	0.05	0.5
za_acc	0.63	0.89	0.91	0.85	0.66	0.66	0.84	0	0.78	0.5	0.8	0.68	0.73	0.65	0.66	0.5	0.79	0.5	0.74	0.99	0.76	0.91	0.94	0.5	0.5	0.29
nad_inst	0.41	0.41	0.99	0.41	0.5	0.5	0.41	0.41	0.41	0.41	0.56	0.83	0.41	0.5	0.5	0.5	0.41	0.41	0.5	0.69	0.5	0.47	0.78	0.5	0.41	0.15
po_acc-R	0.5	0.79	0.99	0.79	0.79	0.99	0.64	0	0.5	0.85	0.5	0.8	0.5	0.3	0.79	0.5	0.86	0.79	0.79	0.79	0.5	0.67	0.89	0.85	0.39	0.52
nom	0.2	0.24	0.43	0.27	0.21	0.52	0.14	0.2	0.22	0.44	0.4	0.34	0.32	0.27	0.5	0.33	0.65	0.18	0.26	0.33	0.4	0.24	0.41	0.31	0.11	0.15
od_gen-R	0.46	0.72	0.59	0.42	0.52	0.45	0.4	0.71	0.74	0.54	0.67	0.53	0.52	0.79	0.74	0.63	0.62	0.48	0.64	0.68	0.68	0.71	0.59	0.52	0.52	0.22
do_gen	0.23	0.47	0.55	0.37	0.57	0.84	0.34	0.52	0.34	0.76	0.77	0.59	0.38	0.45	0.08	0.32	0.85	0.44	0.45	0.63	0.64	0.52	0.67	0.79	0.22	0.18
o_loc	0.48	0.44	0.7	0.47	0.53	0.61	0.46	0.41	0.34	0.5	0.55	0.5	0.27	0.6	0.78	0.56	0.62	0.39	0.52	0.52	0.37	0.48	0.64	0.79	0.41	0.57
na_acc	0.56	0.53	0.61	0.31	0.82	0.23	0.45	0.48	0.5	0.47	0.39	0.48	0.56	0.59	0.63	0.37	0.37	0.59	0.78	0.51	0.56	0.17	0.65	0.69	0.39	0.73
inst	0.44	0.56	0.84	0.21	0.25	0.61	0.49	0.36	0.99	0.7	0.54	0.59	0.15	0.6	0.61	0.72	0.83	0.78	0.4	0.67	0.78	0.41	0.69	0.38	0.32	0.77
acc	0.25	0.51	0.45	0.57	0.44	0.66	0.28	0.47	0.45	0.59	0.55	0.63	0.44	0.69	0.56	0.58	0.77	0.5	0.54	0.68	0.59	0.36	0.63	0.66	0.33	0.41
pod_acc	0.5	0.19	0.5	0.35	0.5	0.5	0.35	0.5	0.5	0.59	0.65	0.53	0.5	0.5	0.5	0.41	0.45	0.5	0.35	0.5	0.61	0.2	0.83	0.5	0.5	0.5
przez_acc-R	0.01	0.01	0.81	0.5	0.9	0.62	0.87	0.58	0.94	0.48	0.99	0.66	0.52	0.54	0.5	0.85	0.98	0.42	0.65	0.95	0.81	0.81	0.77	0.5	0.47	0.24
z_gen	0.16	0.29	0.9	0.52	0.38	0.76	0.22	0.56	0.15	0.58	0.64	0.4	0.32	0.22	0.57	0.43	0.48	0.61	0.38	0.68	0.36	0.47	0.77	0.81	0.32	0.45
po_loc	0.75	0.16	0.68	0.4	0.62	0.5	0.35	0.19	0	0.54	0.72	0.32	0.28	0.24	0.84	0.32	0.54	0.38	0.75	0.5	0.65	0.44	0.79	0.39	0.47	0.6
dat	0.16	0.38	0.16	0.18	0.24	0.49	0.2	0.19	0.94	0.4	0.34	0.75	0.39	0.36	0.5	0.44	0.55	0.21	0.34	0.5	0.46	0.09	0.27	0.5	0.22	0.75
za_inst	0.39	0.49	0.02	0.63	0.5	0.86	0.41	0.14	0.74	0.24	0.65	0.2	0.51	0.97	0.5	0.39	0.88	0.74	0.39	0.94	0.86	0.43	0.67	0.5	0.58	0.5
podczas_gen-R	0.5	0.08	0.5	0.86	0.5	0.5	0.65	0.05	0.5	0.57	0.79	0.77	0.45	0.25	0.5	0.91	0.82	0.5	0.5	0.5	0.99	0.55	0.86	0.87	0.2	0.89
gen	0.1	0.42	0.33	0.33	0.29	0.72	0.36	0.26	0.16	0.57	0.43	0.46	0.51	0.58	0.27	0.71	0.67	0.07	0.4	0.4	0.41	0.32	0.59	0.61	0.82	0.77
przy_loc	0.5	0.44	0.01	0.16	0.5	0.04	0.86	0.5	0.31	0.49	0.51	0.4	0.19	0.23	0.58	0.44	0.94	0.67	0.5	0.5	0.99	0.34	0.75	0.84	0.57	0.5
w_acc-R	0.14	0.06	0.34	0.17	0.15	0.33	0.26	0.14	0.5	0.17	0.26	0.34	0.33	0.45	0.26	0.32	0.17	0.52	0.34	0.21	0.45	0.22	0.29	0.5	0.24	0.73
dla_gen-R	0.08	0.12	0.5	0.01	0.16	0.22	0.14	0.83	0.5	0.65	0.18	0.15	0.69	0	0.5	0.5	0.52	0.9	0.83	0.9	0.5	0.98	0.93	0.5	0.61	0.99

FIGURE 2: Overrepresentation map of the final matrix *syntactic slots* × *noun categories* with grade regression values shown inside the cells (rows and columns ordered by GCA)

Using GCA we optimally ordered the matrix to achieve the maximum positive dependence between columns and rows. The order of categories (columns) becomes now meaningful: the leftmost (motive, communication) and rightmost (time, event, food) columns are now the most dissimilar, with columns between them tending to vary from more similar to (motive, communication), to more similar to (time, event, food).

plant	animal	body	person	object	phenomenon	substance	act	process	cognition	communic.	food	artifact	state	possession	feeling	attribute	event	relation	group	location	motive	quantity	time	shape	TOP
motive	feeling	communic.	cognition	group	person	body	animal	plant	food	artifact	substance	object	phenomenon	shape	attribute	quantity	posses-ion	relation	location	state	process	act	event	time	TOP
motive (0.419)	communic. (0.436)	body (0.441)	attribute (0.449)	feeling (0.456)	animal (0.464)	act (0.469)	state (0.472)	shape (0.481)	group (0.490)	relation (0.499)	artifact (0.502)	process (0.504)	quantity (0.506)	plant (0.512)	phenomenon (0.514)	person (0.516)	TOP (0.519)	possession (0.520)	object (0.528)	substance (0.529)	cognition (0.535)	location (0.542)	food (0.550)	event (0.565)	time (0.571)

FIGURE 3: Two different orderings of semantic categories of noun created by two experts (two upper orderings) and the one obtained automatically by means of grade methods (bottom ordering, with the corresponding grade regression values)

4 Finding distances between noun categories

However, we still work out how to decide whether two or more columns are similar enough to treat them as a group: this depends not only on their final similarity, but on a particular verb and its slot under consideration. Thus, no method based on global clustering of columns in final matrix (cf. Fig. 2) could be applied here. Nevertheless, we can use similarity measures determined by this matrix.

The simplest method is to consider a linear global ordering of *semantic categories* (cf. Fig. 3) as 1D similarity space. For each verb we will consider ordering of *categories*, with respect to frequencies of each category for this particular verb.

Grade regression function values for columns in the final show distances between the columns, so we compute distance in this matrix for pair ⟨motive,time⟩:

$$dist(motive, time) = |Regr_{col}(motive) - Regr_{col}(time)| = |0.571 - 0.419| = 0.152$$

We presuppose that there exist an universal ordering of categories expressing their similarity and it does not depend on particular verbs and syntactic slots. However, the weak regularity of the final matrix (Fig. 2) contradicts this assumption. Moreover, the obtained order differs from the orderings suggested by two experts (Fig. 3). Note that these two hand-made orderings are quite different from each other as well. Thus, the problem tends to be more complicated than we have assumed. To overcome it, we want to apply two other solutions.

The first of them is an extension of algorithms proposed in (**???**). It excludes one by one the most outlying *noun category*. In each step of the procedure we order the matrix by GCA, calculate measure *AvgDist* for each column (the highest is value of *AvgDist*, the more column outlies from others), exclude the most outlying column and repeat the procedure. Therefore we can decide which *category* are so much dissimilar from the others that we cannot combine them with the others, and obtain smaller but presumably more regular set of *categories* that we can later analyse with the first or the second method. Figure 4 shows a plot of *AvgDist* values for *noun categories* in the final matrix.

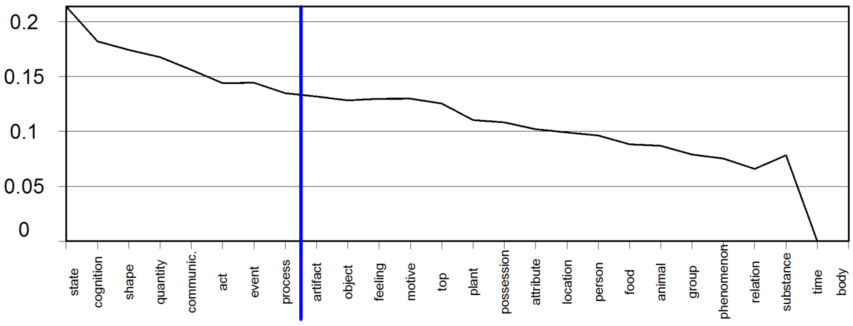


FIGURE 4: Plot of *AvgDist* values, showing how strong each *noun category* was outlying the non-excluded; excluding *categories* started from left (state, which has the highest *AvgDist* value: over 0.2) and step by step each *category* was removed from the matrix

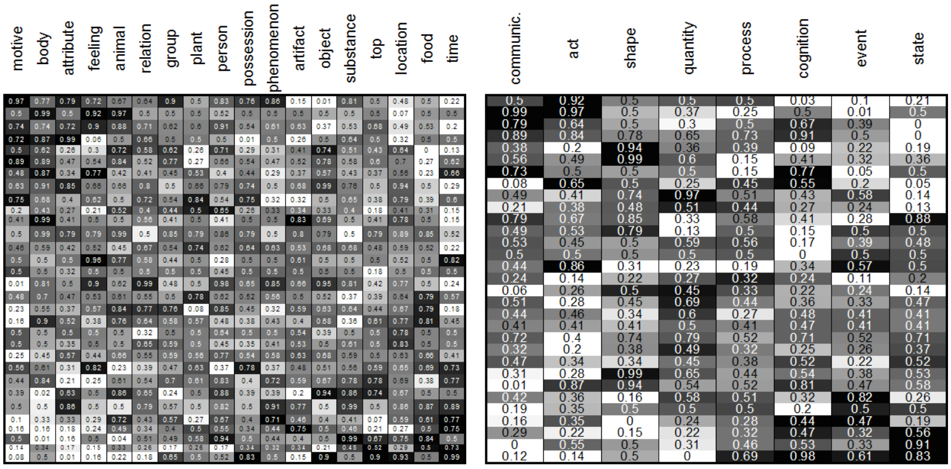


FIGURE 5: Overrepresentation maps of *syntactic slots* × *noun categories* matrix with grade regression values inside (rows and columns ordered by GCA) for 18 "material" (left) and 8 "functional" (right) categories; *syntactic slots* names omitted for clearance

attention! suspicious sentences follows! It seems that excluding outlying categories produces at the beginning set of singletons (the leftmost 8 categories in Fig. 4, separated from the rest by vertical line) weakly related to any other, which would be an obstacle to achieve our goal. Fortunately, we can move these outliers to separate subpopulation and perform grade analysis on a resultant two submatrices. We receive a bigger subpopulation (Fig. 5—left) containing "material" noun categories and a smaller one (Fig. 5—right) containing "functional" categories. Observe that two "inmaterial" categories (shape, quantity, which cannot be interpreted as "functional", land in this subpopulation. Nevertheless, both subpopulations seems to have more sensible linear orderings of categories than the whole final matrix (Fig. 2) covering all 26 categories.

Grade methods provide similarity between every pair of entities (here: semantic categories of nouns) without making it linear. The last method—not covered here because of lack of space—consists in finding connected regions in such a non-linear space by obtaining $ar_{\max}$ index values, which measure the absolute departure between two distribution (here: between two columns, i.e. noun categories).

5 Summary

The main goal of this paper is to try to define what the similarity between *semantic categories of nouns* is (or could be) and, later, to find which *categories* are similar enough to treat them as one group. By using GCA we ordered each independent matrix of *verbs* × *noun categories*, noticed the information on position of every *category* and built the final matrix of *syntactic slots* × *noun categories.* Ordering of this final matrix shows us a trend governing the ordering of rows and columns, and gives us information how distant in 1 dimension (by means of *grade regression values*) every pair of *categories* is.

However, more detailed investigations show that a single linear order of semantic categories is unclear and probably its application to aggregating categories for single slot in semantic valence dictionary would not give satisfactory results. Therefore, we proposed two other methods of determining similarity between them that should overcome these shortcomings. Nevertheless, the quality of the obtained orderings could be verified only in a process of creation of the entire dictionary.

The final judging of quality of each method is performing it on small set of verbs, so the experts will be able to decide which aggregation of *noun categories* for each verb is the most meaningful and correct. This is our task for the future work.